I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail, in an envelope addressed to: MS Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below.

Dated: 9/24/07 Signature: Maureen DiVito
(Maureen Divito)

Docket No.: 0081004.00167US2 (RSA-044)
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Ari JUELS

Application No.: 09/802,278

Confirmation No.: 6866

Filed: March 8, 2001

Art Unit: 3621

For: TARGETED DELIVERY OF INFORMATIONAL CONTENT WITH PRIVACY PROTECTION

Examiner: Elisca, Pierre E.

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

DECLARATION UNDER 37 C.F.R. §1.131

I, Ari Juels, hereby declare as follows:

1. I am the inventor of the above-referenced patent application.

2. All the work described within this declaration was performed in the United States.

3. All of the work described within this declaration was performed by me, or on my behalf and under my direction.

4. I have reviewed our records, including the exhibits submitted herewith, and readily declare that a method for enabling targeted information retrieval while protecting consumer privacy, as claimed within the subject application, including original claims 1-23, was conceived at least prior to February 29, 2000, i.e., the priority date of U.S. Patent Application US 2001/0049620 to Blasko.

5. Attached is a document entitled "Targeted Adverstising . . . And Privacy Too" (adprivacy14Feb00.pdf), completed at least by February 14, 2000, describing the claimed invention. This document was prepared by me, and describes a simple, practical technical solution that enables sophisticated use of detailed consumer profiles for the purposes of targeting advertisements, but protects these profiles from disclosure to advertisers or hostile third parties.

6. This document clearly indicates that that the invention comprising a method for enabling targeted information retrieval while protecting consumer privacy, as claimed within the subject application, including present claims 1- 23, was conceived at least as early as February 14, 2000.

7. Also attached are the following documents, prepared by me:

A. "Targeted Adverstising . . . And Privacy Too" (adprivacy17Feb00.pdf), completed at least by February 17, 2000.

B. "Targeted Adverstising . . . And Privacy Too" (adpriv1.pdf), completed at least by February 26, 2000.

8. The documents listed in paragraph 7 above demonstrate diligence from the date of conception to at least February 26, 2000.

9. The Provisional Patent Application 60/187,671, to which the present patent application claims priority, was filed on March 8, 2000.

10. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that the making of willfully false statements and the like is punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signed: Ari Juels
Ari Juels

Dated: 21 Sept. 2007

adpriv1.pdf

Targeted Advertising... And Privacy Too

Ari Juels[1]

RSA Laboratories
Bedford, MA 01730, USA
E-mail: ajuels@rsasecurity.com

Abstract.

1 Introduction

In February 2000, a major Web advertising firm known as DoubleClick touched off a furore in the press with the announcement of a more aggressive policy of consumer data aggregation. DoubleClick declared that it would begin to integrate offline information about consumers into its existing database of online information derived from surveillance of consumer Web surfing [?]. This announcement came in the midst of a number of articles in the popular press regarding surreptitious sharing of consumer information. A week earlier, a report released by the California HealthCare Foundation alleged that a number of health-related Web sites were violating their own stated privacy policies and divulging sensitive information about customers to third parties [?]. The next day, Reuters reported that two suits for privacy invasion and an investigation by the Federal Trade Commission were pending against Amazon.com and its subsidiary Alexa [?]. While consumer and privacy advocacy groups vigorously decry such abuses, advertisers defend their policy of harvesting and exploiting demographic information by highlighting the benefits of targeted advertising. Consumers, they maintain, are more likely to find interest in advertising tailored to their own preferences, and such advertising consequently leads to greater consumer market efficiency. The United States government has addressed the issue by promoting a policy of industry self-regulation, leading to friction with the European Union, which has sought more stringent consumer privacy guarantees.

In this paper, we show that targeted advertising and consumer privacy need not in fact be conflicting aims. We describe a simple, practical technical solution that enables use of detailed consumer profiles for the purposes of targeting advertisements, but protects these profiles from disclosure to advertisers or hostile third parties. Somewhat surprisingly, the most basic embodiments of our idea do not even require use of cryptography.

The underlying idea is quite simple. Rather than gathering information about a consumer in order to decide which advertisements to send her, an advertiser makes use of a client-side software module called a *negotiant*. The *negotiant* serves a dual purpose: It acts as a client-side proxy to protect user information, and it also directs the targeting of advertisements. The negotiant requests advertisements from the advertiser that are tailored to the profile provided by the user. The advertiser can control the palette of advertisements available to the negotiant, as well as the process by which it decides which ads to request. At the same time, the advertiser learns no information about the consumer profile beyond which advertisements the negotiant requested. In more sophisticated variants, the negotiant is able to participate in a protocol whereby the advertiser does not even learn what ads a given user has requested, but only sees ad requests in the aggregate. The end result is that the advertiser is able to target ads with a high degree of sophistication, and also to gather information on ad display rates, all without learning significant information about individual consumer profiles.

Of course, some restriction must be placed on advertiser control of negotiants. Otherwise, the advertiser can manipulate them so as to extract detailed profile information from individual consumers. The fact that negotiants may be viewed and controlled by users helps offset this vulnerability, as we discuss in the body of the paper. An additional point of concern in our proposal is a restriction that it places on advertisers. With the use of negotiants, advertisers cannot correlate profile information among users, as is possible when consumer profiles are collected in a central location. This drawback should be partially offset by the fact that the negotiant may safely and privately gain access to a great deal of sensitive information that would otherwise not be available to advertisers. Nonetheless, we propose some strategies for addressing this limitation in a manner that preserves consumer privacy.

1.1 Previous Work

A negotiant may be viewed as a client-side software proxy. The related approach of using server proxies as a means of protecting consumer privacy is a well established one, and has even produced a number of commercial ventures. For a small subscription fee, companies such as Zero-Knowledge Systems [?] offer customers an encrypted channel to one or more proxy servers that anonymously reroute requests to destination servers. The proxy servers thus act as intermediaries between the client and Web servers, shielding the client from positive identification. The client, of course, must trust at least one of the servers to ensure her anonymity, and must also trust servers not to eavesdrop on or tamper with her communications. (Encryption or authentication, where available, may alleviate this latter problem.) Proxy services may be cryptographically strengthened through the use of *mix networks*. A mix network is essentially a distributed cryptographic algorithm for interleaving multiple channels so as to anonymize them. We describe the idea in more detail in Section 2.1. For on-the-fly communications, however, mix networks are often not practical, and therefore not yet employed in real-world applications.

A variant on the idea of proxy servers is the Crowds project at AT&T Labs [?,?,?]. A "crowd" is a group of users, preferably with disparate geographical and other characteristics, that serve to shield one another's identities. The service requests of a user in a crowd are randomly rerouted through other crowd members, rendering the identity of the user indistinguishable from those of other crowd members. In this system, trust is embodied partly in an administrative server responsible for forming crowds, and partly in other crowd members. In particular, the user trusts other crowd members not to eavesdrop on or tamper with communications, and, to a lesser extent, not to perform traffic analysis.

The proxy server and crowd approaches seek to provide a maximum of consumer privacy. While they can be combined with cookies, or other user tracking devices, they do not aim to accommodate more fine-grained control of Web server access to user data. The Platform for Privacy Preferences Project, known as P3P [?], focuses precisely on this latter problem of refined user control of personal demographic information. The goal of P3P is to enable Web sites to publish precise specifications of their privacy policies, and to enable users to exercise control over their private data in response to these policies. The platform allows users to define preferences over what data they are willing to divulge, and also policies about how to respond to site practices that are incompatible with their preferences. Under the aegis of the World Wide Web (W3) Consortium, P3P aims to set forth a standard syntax and body of protocols for general use on the Web. An initial version of the standard is close to completion, a "last call" specification having been released for public review in November 1999.

Underlying the P3P approach is the presumption that mediation between consumers and advertisers is a matter of deciding what information consumers choose to reveal explicitly. As we explain above, we set forth a different approach in which consumers and advertisers to decide jointly in a privacy protecting manner what advertisements consumers should be provided with. There are a number of cryptographic tools that enable theoretically strong privacy protection in this approach. For the more strongly privacy protecting variants of our negotiant scheme, we consider variants on the idea of *private information retrieval* (PIR). A PIR scheme enables a client to request a piece of data from a server – such as an advertisement – in such a way that the server learns no information about the client request.

More formally, let Bob be a user, and let Alice be a server that maintains a database containing items $\alpha = \{a_1, a_2, \ldots, a_n\}$. Alice might represent an advertiser, and α might represent the collection of advertisements held by Alice. The aim of a PIR scheme is to enable Bob to retrive an element $a_r \in \alpha$ of his choice from Alice in such a way that Alice learns no information about r. Of course, this may be accomplished trivially by having Alice send all of α to Bob. As shown in [?], however, a single-server PIR scheme may in fact be designed with $o(n)$ communication, in particular, n^ϵ communication for any $\epsilon > 0$ under the quadratic residuosity assumption. This was recently improved to $polylog(n)$ communication overhead under the so-called phi-hiding assumption [?]. A number of variant PIR schemes have been proposed in the literature, such as symmetric PIR (SPIR) schemes, which include the additional property that the client sees only the data it has requested [?], and a variant with auxiliary servers [?]. None of these proposed PIR schemes, however, is practical for wide scale deployment.

In this paper, we consider a practical alternative to these proposed PIR schemes. In order to obtain improved communications and computational efficiency, we consider two relaxations of the common security model. First, in lieu of a single server (Alice), we assume a collection of servers among which a majority behave in an honest fashion. We refer to this as a *threshold* PIR scheme. As we show, a threshold PIR scheme is capable of achieving communication overhead of $\Theta(1)$ per consumer request under appropriate computational assumptions. As a second, additional assumption, we consider a scenario in which requests from a large number of users may be batched, in which case it is acceptable for servers to learn what has been requested, but not by whom. In other words, in consonance with the Crowds principle, we permit full disclosure of aggregate information, but hide information regarding the requests of individual users. We refer to a threshold PIR scheme with this latter property as a *semi-private* PIR scheme. A semi-private PIR scheme, in addition to achieving communication overhead of $\Theta(1)$, is computationally quite efficient, involving $\Theta(1)$ basic cryptographic operations per item per server.

The negotiant approach we propose in this paper is not necessarily meant as a substitute for proxy servers, Crowds, or P3P. It may instead be viewed as a complementary technology, deployable in conjunction with any of these other ideas. For example, a user might use a proxy server and client-side negotiant function together, or might provide demographic information to a trusted server, allowing it to make use of a negotiant function. Moreover, any of a range of tradeoffs between efficiency and security may be used in the construction of a negotiant function. We show this by presenting in this paper not one, but four different negotiant schemes.

1.2 Organization

In Section 2, we describe the basic cryptographic primitives used in our more advanced negotiant protocols. We also formalize the model in which we propose our schemes, and set forth basic definitions regarding privacy. In Section 3, we propose some negotiant function constructions. We consider some practical implementation issues in Section 4, and conclude in Section 5 with a brief discussion of some future avenues of investigation.

2 Preliminaries

2.1 Building blocks

Let us begin by introducing some of the cryptographic primitives used in the more advanced variants of our protocol. Readers familiar with the basic cryptographic literature may wish to skip to Section 2.2. Most of the protocols we describe are (j, m)-*threshold* protocols. These are protocols executed by a collection of servers $S_1, S_2, \ldots, S_m$, where $m \geq 1$, such that protocol privacy and the correctness of the output are ensured given an honest coalition of any j servers. In such protocols, servers hold a private key x in an appropriate distributed fashion, with a corresponding published public key $y = g^x$. It is common to use the Pedersen protocol [?,?] as a basis for distributed key generation, although see [?] for a caveat. We do not discuss key generation or any of the other ordinary elements of distributed discrete log cryptographic algorithms in detail, but instead refer the reader to, e.g., [?] for a survey, or to any of the papers we reference with regard to specific algorithms.

El Gamal cryptosystem: Where we require public key cryptography in our schemes, we employ the El Gamal cryptosystem [?,?]. Encryption in this scheme takes place over a group G_q of prime order q. Typically, G_q is taken to be a subgroup of Z_p^*, where $q|(p-1)$. Alternatives are possible; for example, G_q may be the group of points of an elliptic curve over a finite field.[1]

Let g be a generator of G_q. This generator is typically regarded as a system parameter, since it may be used in multiple key pairs. The private encryption key consists of an integer $x \in_U Z_q$, where $\in_U$ denotes uniform random selection. The corresponding public key is defined to be $y = g^x$. To encrypt a message $M \in G_q$, the sender selects $z \in_U Z_q$, and computes the ciphertext $(\alpha, \beta) = (My^z, g^z)$. To decrypt this ciphertext using the private key x, the receiver computes $\alpha/\beta^x = My^z/(g^z)^x = M$. We assume a consistent choice of g as a generator for all instantiations of the El Gamal cryptosystem in this paper.

The El Gamal cryptosystem is *semantically secure* under the Decision Diffie-Hellman assumption over G_q [?]. Informally, this means that an attacker who selects message pair (m_0, m_1) is unable to distinguish between encryptions of these two messages with probability non-negligibly greater than 1/2. See, e.g., [?] for futher details.

Let $(\alpha_0, \beta_0) \otimes (\alpha_1, \beta_1) \equiv (\alpha_0\alpha_1, \beta_0\beta_1)$. Another useful property of the El Gamal cryptosystem is the fact that it possesses a *homomorphism* under the operator $\otimes$. In particular, observe that if (α_0, β_0) and (α_1, β_1) represent ciphertexts corresponding to plaintexts M_0 and M_1 respectively, then $(\alpha_0, \beta_0) \otimes (\alpha_1, \beta_1)$ represents an encryption of the plaintext $M_0 M_1$. A consequence of this homomorphic property is that it is possible, using knowledge of the public key alone, to derive a random *re-encryption* (α', β') of a given ciphertext (α, β). This is accomplished by computing $(\alpha', \beta') = (\alpha, \beta) \otimes (\gamma, \delta)$, where (γ, δ) represents an encryption of the plaintext value 1. It is possible to prove quite efficiently in zero-knowledge that (α', β') represents a valid re-encryption of (α, β) using, e.g., a variant of the Schnorr proof of knowledge protocol [?]. This proof may also be made non-interactive. See [?] for an overview.

Quorum controlled asymmetric proxy re-encryption: This is a threshold algorithm enabling re-encryption of an El Gamal ciphertext under a new key. Input to

[1] Most commonly, we let $p = 2q + 1$, and we let G_q be the set of quadratic residues in Z_p^*. In this setting, plaintexts not in $\mathcal{G}_q$ can be mapped onto $\mathcal{G}_q$ by appropriate forcing of the LeGendre symbol, e.g., inversion of the associated integer sign.

the protocol is an El Gamal public key y', as well as a ciphertext $(\alpha, \beta) = E_y[M]$. The output of the protocol is $(\alpha', \beta') = E_{y'}[M]$. Jakobsson proposes a protocol that is computationally secure in the sense that it is robust against any adversary controlling any minority coalition of cheating servers, and also preserves the privacy against such an adversary. Additionally, the protocol is efficient in a practical sense. See [?] for details.

Distributed plaintext equality test This is a threshold protocol whereby, given El Gamal ciphertexts (α, β) and (α', β'), a collection of servers determine whether the underlying plaintexts are identical. The basic idea is that each server blinds the ciphertext by raising α and β to a random exponent, and then proves the blinding correct. The resulting blinded ciphertext is then decrypted, yielding a 1 if the underlying plaintexts are equivalent, and a random value otherwise. See [?] for an efficient and practical protocol construction and proofs of security. We write $(\alpha, \beta) \approx (\alpha', \beta')$ to denote equality of underlying plaintexts.

Bulletin Board: Our proposed schemes with multiple players or servers assume the availability of a *bulletin board*. This may be viewed as a piece of memory which any player may view and to which all players have appendative write access. A bulletin board may be realized as a public broadcast channel, or is achievable through Byzantine agreement or some appropriate physical assumption. See [?,?] for description of a practical implementation of this primitive. Postings to a bulletin board may be made authenticable, i.e., their source may be securely validated, through use of such mechanisms as digital signatures. In many cases, our proposed algorithms only require bulletin board access by servers, not by other players.

Mix networks: A critical building block in our protocols is a threshold algorithm known as a *mix network*. Let $E_y[M]$ represent the encryption under public key y of message M in a probabilistic public-key cryptosystem, typically El Gamal. This notation is informal, in the sense that it does not take into account the random encryption exponent that causes two encryptions of the same plaintext to appear different from one another. While we retain this notation for simplicity, the reader must bear it in mind, particularly with regard to the fact that mix networks involve re-encryption of ciphertexts.

A mix network takes as input a vector of ciphertexts $V = \{E_y[M_1], E_y[M_2], \ldots, E_y[M_n]\}$. Output from the mix network is the vector $V' = \{E_y[M_{\sigma(1)}], E_y[M_{\sigma(2)}], \ldots, E_y[M_{\sigma(n)}]\}$, where σ is a random permutation on n elements. A mix scheme is said to be *robust* if, given a static adversary with active control of a minority coalition of servers, V' represents a valid permutation and re-encryption of ciphertexts in V with overwhelming probability. A mix scheme is said to be *private* if, given valid output V', for any $i \in \{1, 2, \ldots, n\}$, an adversary with active control of a minority coalition and passive control of at most $m-1$ servers cannot determine $\sigma^{-1}(i)$ with probability non-negligibly larger than $1/n$. It should be noted that to prevent attacks in which some players post re-encryptions of other players' inputs, it is often a requirement that input be encrypted in a manner that is *plaintext aware*. For this, it suffices that a player presenting El Gamal ciphertext (α, β) also provide a zero knowledge proof of knowledge of $\log_g \beta$, and that servers check the correctness of this proof. See [?] for further details.

Mix servers were introduced by Chaum [?] as a basic primative for privacy. In his simple formulation, each server S_i takes the output V_i of the previous server and simply permutes and re-encrypts the ciphertexts therein. While this scheme is private, it is not robust. Robust, threshold mix networks were introduced in [?,?]. The most efficient mix network to date is the construction of Jakobsson, which requires $O(n)$ computation and communication per server. The scheme makes use of the El Gamal cryptosystem. For further details and formal definitions, the reader is referred to [?]. Given its robustness and efficiency on large batches, the Jakobsson construction is probably most appropriate for our schemes. It will be observed, however, that robustness is not of critical importance in these schemes, as a server corrupting the computation can at best insert a false or incorrectly advertisement, something likely to be detected if widespread. Hence, even the mix network proposed by Chaum may often be appropriate for our proposed schemes.

There are many variations on mix networks. For example, there are efficient mix networks in which V is a vector of k-tuples of ciphertexts. Additionally, a mix network may take either ciphertexts or plaintexts as inputs and likewise output either plaintexts or ciphertexts. We employ a variety of such operations in our protocols, and do not describe implementation details.

2.2 Model and definitions for our scheme

Let $C_1, C_2, \ldots, C_k$ be a collection of consumers toward whom advertisements are to be directed. Let $P_1, P_2, \ldots, P_k$ be the respective profiles of these consumers. These profiles may contain any of a variety of pieces of information on the consumer, including standard demographic information such as age, sex, profession, annual income, etc., as well as other information such as recently visited URLs and search engine queries. Let us designate the set of possible consumer profiles by $\mathcal{P}$. We denote the advertiser by A, and let $\alpha = \{a_1, a_2, \ldots, a_n\}$ be the set of advertisements that A seeks to distribute. The advertiser chooses a *negotiant function* $f_\alpha : \mathcal{P} \rightarrow Z_n$. This function takes the profile of a consumer as input and outputs a choice of advertisement from among α. (We shall write f for clarity of notation, leaving the subscript implicit.) As an example, f might derive a list of the most common words in URLs visited by the user and seek to match these to descriptors associated with the ads in α. Of course, it is possible to include any of a wide variety of additional inputs to f, such as the current date, or the list of advertisements already sent to the consumer. We assume that A is represented by a set of servers $S_1, S_2, \ldots, S_m$, for $m \geq 1$; these servers share a bulletin board. All consumers post ad requests to the bulletin board. Servers then initiate communication with consumers and dispense ads to them. The following is a list of definitions and properties useful in describing negotiant protocols.

Let l be an appropriately defined security parameter. We say that a function $q(l)$ is *negligible* if for any polynomial p, there exists a value d such that for $l \geq d$, we have $q(l) < 1/|p(l)|$. Otherwise, we say that q is *non-negligible.* We say that probability $q(l)$ is *overwhelming* if $1 - q(l)$ is negligible.

Let A_1 be a polynomial-time adversary that actively controls a static minority set of servers, or, if there is only one server, controls that single server. In other words, let us suppose that A_1 controls $max(\lfloor m/2 \rfloor, 1)$ servers. In addition, let us suppose that A_1 knows f and α. Consider the following experiment. We assume without loss of generality that A_1 does not control consumer C_1. A_1 chooses a pair of profiles $\tilde{P}_0, \tilde{P}_1 \in \mathcal{P}$. A bit $b \in \{0, 1\}$ is selected at random and P_1 is set to $\tilde{P}_b$. Now the protocol is executed, and A_1 outputs a guess for $b¿$ We say that the protocol has *full privacy* if for any adversary A_1, it is the case that pr$[A_1$ outputs $b] - 1/2$ is negligible, where the probability is taken over the coin flips of all participants. This definition states informally that the protocol transcript reveals no significant information about P_1, even if all other consumers are in the control of A_1.

Now let us modify the experiment slightly and consider a polynomial-time algorithm A_2 that is provided only with the input $f(P_1), f(P_2), \ldots, f(P_k)$ in a random order, as well as f and α. We say that a negotiant protocol has *profile privacy* if for any A_1 and any $P_2, P_3, \ldots, P_k$, there exists an A_2 such that pr$[A_1$ outputs $b] -$ pr$[A_2$ outputs $b]$ is negligible if all consumers execute the protocol correctly. Again, probabilities are taken over the coin flips of all participants. In other words, the protocol transcript reveals no signficant information about P_1 other than that revealed by the aggregate ad requests of the participating consumers. Note that when $m = 1$, the property of profile privacy means that an advertiser learns only the ad requests of a given consumer. When $m > 1$, the property implies that an advertiser learns only the aggregate ad requests of a group of consumers.

We say that a negotiant protocol is *aggregate transparent* if any server can determine $f(P_1), f(P_2), \ldots, f(P_k)$ with overwhelming probability. In real-world advertising scenarios, it is important that a protocol be aggregate transparent, as the clients of advertisers typically wish to know how many times their ads have been displayed.

3 Some Negotiant Schemes

We now present several negotiant schemes representing a small spectrum of tradeoffs between security properties and resource costs.

3.1 Scheme 1: Naive PIR scheme

We present this simple scheme as a conceptual introduction. Here, requests are directed from a single consumer C with profile P to a single server S. (Thus the scheme may be modeled by $m = k = 1$.) The scheme is this: The server sends all of α to C, who then views ad $a_{f(P)}$.

Clearly, this scheme enjoys full privacy. The chief drawback is the $O(n)$ communication cost. Another drawback is the fact that the scheme is not aggregate transparent. Nonetheless, given a limited base of advertisements and good bandwidth, and if advertisers are satisfied with recording click-through rates, this scheme may be useable in certain practical scenarios.

3.2 Scheme 2: Direct request scheme

This is another conceptually simple scheme involving a one-on-one consumer and server interaction. In this scheme, C simply sends $f(P)$ to S, who returns $a_{f(P)}$. This scheme enjoys profile privacy and has communication and computation costs $\Theta(1)$. Despite (or because of) its simplicity, it is from a practical standpoint in fact quite appealing.

3.3 Scheme 3: Semi-private PIR scheme

We now show how to invoke some of the cryptographic apparatus described above in order to achieve a semi-private PIR scheme useable as the basis for a negotiant scheme. Given database $\alpha = \{a_1, a_2, \ldots, a_n\}$, the goal is for a collection of consumers $C_1, C_2, \ldots, C_k$ to retrieve respective elements $a_{r_1}, a_{r_2}, \ldots, a_{r_k}$ in such a way that the database servers learn requests only in the aggregate. Of course, our aim here is to apply this scheme to the retrieval of advertisements, and we shall present it in this context. In other words, we assume that $r_i = f(P_i)$, i.e., users are consumers seeking to retrieve advertisements. As above, we assume a public/private El Gamal key pair (y, x) held in an appropriate distributed manner by servers $S_1, S_2, \ldots, S_m$. We also assume that each consumer C_i has a public/private El Gamal key pair (y_{C_i}, x_{C_i}). The scheme is as follows.

1. Each consumer C_i computes $r_i = f(P_i)$ and posts the pair $(E_y[r_i], i)$ to the bulletin board. Let $V_1 = \{E_y[r_i], i\}_{i=1}^{k}$ be a vector of ciphertext/plaintext pairs accumulated when all consumers have posted their requests.
2. Servers apply a mix network to V_1 to obtain V_2, where V_2 is a vector of pairs $\{(r_{\sigma_1(i)}, E_y[\sigma_1(i)])\}_{i=1}^{k}$ for random, secret permutation σ_1.
3. Servers replace each integer r_j in V_2 with a_{r_j}. Call the resulting vector V_2'.
4. Servers apply a mix network to V_2' to obtain a vector V_3, where V_3 is a vector of pairs $\{(E_y[a_{r_{\sigma_2(i)}}], \sigma_2(i))\}_{i=1}^{k}$, and σ_2 is an random, secret permutation.
5. Let $(E_y[a_{r_i}], i)$ be an element in V_3. For each pair, the servers apply quorum controlled asymmetric proxy re-encryption to obtain $(E_{y_{C_i}}[a_{r_i}], i)$. Let the resulting vector be V_4.
6. For each element $(E_{y_{C_i}}[a_{r_i}], i)$ in V_4, servers send $E_{y_{C_i}}[a_{r_i}]$ to C_i.
7. Consumers decrypt their respective ciphertexts.

The security of the scheme is predicated on that of the underlying mix network. If we use that proposed in [?], for example, it may be shown that this is a semi-private PIR scheme, i.e., enjoys profile privacy, relative to the Decision Diffie-Hellman assumption. Assuming that a public key operation in G_q incurs cost $\Theta(l^3)$, where the security parameter l is linearly related to the bit length $|q|$, the computational costs of the scheme are $\Theta(l^3)$ per element per server. The communication costs of the scheme are $\Theta(1)$. With appropriate implementation enhancements, some of which we discuss in Section 4, we believe that this scheme may be deployed in a practical manner.

3.4 Scheme 4: Threshold PIR

The semi-private PIR scheme described above can be converted into a threshold PIR scheme with a few extra steps, and at the expense of additional computational overhead. The idea is to perform a blind lookup of consumer ad requests. This is accomplished by mixing ads and then invoking the distributed plaintext equality test described in Section 2.1. The construction is such that processing consumer requests one at a time is as efficient as processing many simultaneously. We therefore present the protocol as applied to a single consumer C with profile P and private/public key pair (y_C, x_C). Consumer C computes $r = f(P)$ and posts $E_y[r]$ to the bulletin board. The protocol is then as follows.

1. Servers construct a vector U_1 of pairs $\{(j, E_y[a_j])\}_{j=1}^{n}$.
2. Servers mix U_1 to obtain a vector U_2 of the form $(E_y[\sigma(j)], E_y[a_{\sigma(j)}])$ for a random, secret permutation σ.
3. For each j, the servers perform a distributed plaintext equality test to see whether $E_y[j] \approx E_y[r]$. Assuming correct protocol execution, when a match is found, this yields the ciphertext pair $(E_y[r], E_y[a_r])$.
4. The servers apply quorum controlled asymmetric proxy re-encryption to obtain $E_{y_C}[a_r]$. They send this to C.
5. C decrypts $E_{y_C}[a_r]$ to obtain a_r.

This protocol has communication complexity $\Theta(1)$. The computational complexity is $O(nl^3)$ per server.

4 Security and Implementation Issues

4.1 Attacks outside the model

We have offered cryptographically based characterizations of the security of our schemes, showing their adherence to the appropriate definitions in 2.2. In particular, we see that for Schemes 2 and 3 that an attacker in control of a minority coalition of servers can learn little beyond individual or aggregate ad requests. As mentioned above, however, even with these security guarantees an advertiser with full control of the negotiant function f can manipulate it so as to extract detailed profile information from individual users. Let us suppose, for example, that an advertiser wishes, through Scheme 2, to learn the approximate annual household income in dollars of a given consumer C with profile P. The advertiser can, for example, construct a function f such that $f(P) = \lfloor I/10,000 \rfloor$, where I is the annual household income of the consumer. In fact, given enough latitutude in the distribution of the negotiant function to consumers, an advertiser can even defeat the aggregate security of Scheme 3, by distributing a different function to each consumer. We propose a number of possible safeguards against such abuses.

- **Open source negotiant function:** The idea here is to allow easy reverse engineering of f by consumers or watchdog organizations. This may be accomplished by requiring that f be encoded in a high level language, and even providing software tools for dissecting it. Consumers or organizations that deem f unduly invasive may refuse to receive advertisements or lodge complaints against the advertiser, as appropriate.
- If f is unduly obscured or intrusive, consumers may complain against or boycott the advertiser. To ensure that f is consistent from user to user, a signed and timestamped hash should be publicly posted, or else f should be distributed by some trusted site

4.2 Mixing need not be on-the-fly

User can request ad r, and have it returned at some later time.

4.3 Implementation efficiency enhancements

Bulk encryption We assume in Schemes 3 and 4 that an advertisement may be represented as a single ciphertext. Of course, in reality, it is impractical to use ads small enough or a group G_q large enough to support this assumption. We may, however, represent an advertisement as a sequence of associated ciphertexts. Alternatively, we may use an enveloping scheme, and represent a encryption of M as $\tilde{E}_y[M] = (\gamma, \delta)$, where $\gamma = \{E_y[\kappa_1], E_y[\kappa_2], \ldots, E_y[\kappa_z]\}$ and $\delta = \epsilon_{\kappa_z}\epsilon_{\kappa_{z-1}}\epsilon_1[M]$. Here, $\epsilon_\kappa[M]$ represents a symmetric-key encryption of M, where $\kappa \in K$ is a key from keyspace K. To re-encrypt $\tilde{E}_y[M]$ as (γ', δ'), a player does the following:

1. Re-encrypt all ciphertexts in γ.
2. Select $\kappa_{z+1} \in_U K$.
3. Append $E_y[\kappa_{z+1}]$ to γ to obtain γ'.
4. Compute δ' as $\epsilon_{\kappa_{z+1}}[\delta]$.

There are two major drawbacks to this scheme. First, the size of a ciphertext, as well as the computational cost of re-encryption, grows linearly in the number of re-encryptions z. While this leads to poor asymptotic performance, it may substantially enhance the performance in practice, particularly when m is small and ad sizes are large. A second drawback is that it is unclear how to achieve robustness in an efficient way in a mix network employing such a scheme. We submit, however, that robustness is a less important consideration than privacy in our schemes.

Reducing public-key operation costs It should be noted that all of the costly operations in our scheme involve exponentiations in G_q. These may be made quite inexpensive through the use of pre-processing or addition chains. Thus, assuming 100,000 elements, the total cost is roughly that of...

Dishonest behavior among consumers Adversary control of consumers

4.4 Construction of f

4.5 Correlated consumer data

Advertisers are often interested in obtaining aggregate demongraphic information, or in determining correlations among various data items. We propose two possible means of accomplishing this. First, it is possible to allow consumers to sell their demographic information in exchange for money or services. (This is effectively the contract behind free ISPs such as NetZero.) This of course will present a somewhat biased picture. Another possible means is to invoke a mix network

5 Conclusion

This paper seeks to convey two ideas, the first cryptographic and the second political. On the cryptographic side, we observe that by relaxing certain of the security assumptions surrounding the conventional PIR model, we are able to achieve considerable practical improvements in terms of both communication and computational complexity. While the privacy guarantees offered by the schemes we propose are somewhat weaker than theory allows, they are often quite adequate for real-world applications. On the political side, we have offer a new perspective on the contention between online advertisers and consumer privacy advocates. We demonstrate a conceptually simple technical approach to advertising that brings the objectives of both camps into closer alignment.

Of the farrago of issues and problems we leave unaddressed, we conclude by mentioning some here as topics for future research. Perhaps the most pressing, and also the most complex, is how to harmonize the numerous approaches to privacy and consumer convenience in the literature and in practice. For

Our proposed threshold PIR scheme, i.e., Scheme 4, raises an interesting cryptographic question. In that scheme, it is as efficient to process consumer requests one at a time as it is to process them in a batch. Is there some form of batch processing that can improve the efficiency of this scheme without a significant weakening of the underlying security model?

One important issue is that of server cookies. In many cases, consumers are quite willing to accept cookies as a means of customizing or simplifying their experience of a Web site. For example, customers visiting the Amazon.com site may wish to use cookies in order to facilitate the provision of book recommendations and to make the purchase process easier. In this case, as users often provide credit card and address information, their use of cookies is essentially a means of self-identification, although it does not leak extensive demongraphic information directly. Regulation of information gathered through distribution of these cookies is an important legislative question.

adprivacy14Feb00.pdf

Targeted Advertising...
And Privacy Too

Ari Juels[1]

RSA Laboratories
Bedford, MA 01730, USA
E-mail: ajuels@rsasecurity.com

Abstract.

1 Introduction

In February 2000, a major Web advertising firm known as DoubleClick touched off a furore in the press with the announcement that it would integrate offline information about consumers into its existing database of online information derived from surveillance of consumer Web surfing. This announcement by DoubleClick came on the heels of a number of other articles in the popular press regarding abuse of private consumer information. A week earlier, a report released by the California HealthCare Foundation alleged that a number of health-related Web sites were violating their own stated privacy policies and divulging sensitive information about customers to third parties. The next day, Reuters reported that two suits for privacy invasion and an investigation by the Federal Trade Commission were pending against Amazon.com and its subsidiary Alexa. While consumer and privacy advocacy groups vigorously decry such abuses, advertisers defend their policy of harvesting and exploiting demographic information by highlighting the benefits of targeted advertising. Consumers, they maintain, are more likely to find interest in advertising tailored to their own preferences, and such advertising consequently leads to greater consumer market efficiency. The United States government has addresed the issue by promoting a policy of industry self-regulation, leading to friction with the European Union, which has sought more stringent consumer privacy guarantees.

In this paper, we show that targeted advertising and consumer privacy need not in fact be conflicting aims. We describe a simple, practical technical solution that enables sophisticated use of detailed consumer profiles for the purposes of targeting advertisements, but protects these profiles from disclosure to advertisers or hostile third parties. Somewhat surprisingly, the most basic embodiments of our idea do not even require use of cryptographic techniques.

The underlying idea is quite simple. Rather than gathering information about a consumer in order to decide which advertisements to send her, an advertiser makes use of a client-side software module called a *negociant.* The *negociant* serves a dual purpose: It acts as a client-side proxy to protect user information, and also directs the targeting of advertisements. The negociant requests advertisements from the advertiser that are tailored to the profile provided by the user. The advertiser can control the palette of advertisements available to the negociant, as well as the process by which it decides which ads to request. At the same time, the advertiser learns no information about the consumer profile beyond which advertisements the negociant requested. In more sophisticated variants, the negociant is able to participate in a protocol whereby the advertiser does not even learn what ads a given user has requested, but only sees ad requests in the aggregate. The end result is that the advertiser is able to target ads with a high degree of sophistication, and also to gather information on ad display rates, all without learning significant information about individual consumer profiles.

Of course, some restriction must be placed on advertiser control of negociants. Otherwise, the advertiser can manipulate them so as to extract detailed profile information from indivudal consumers. The fact that negociants may be viewed and controlled by users helps offset this vulnerability, as we discuss in the body of the paper. An additional point of concern in our proposal is a restriction that it places on advertisers. With the use of negociants, advertisers cannot correlate profile information among users, as is possible when consumer profiles are collected in a central location. This drawback should be partially offset by the fact that the negociant may safely and privately gain access to a great deal of sensitive information that would otherwise not be available to advertisers. Nonetheless, we propose some strategies for addressing this limitation in a manner that preserves consumer privacy.

1.1 Previous Work

A negociant may be viewed as a client-side software proxy. The approach of using proxies as a means of protecting consumer privacy is a well established one, and has seen application by both research and commercial ventures. These efforts, however, have typically involved proxy *servers*, i.e., servers that act as intermediaries between consumers and Web sites. (Of course, it is possible to install a negociant on a proxy server, rather than the client. This requires, however, an embodiment of trust in the proxy server.) A number of proposals...

Consumer privacy in P3P – language of privacy

In the stronger variant of our scheme, we wish to ensure against advertisers learning which advertisements have been requested by which users. In principle, it is possible for a consumer to request an advertisement from a server such that the server learns no information about the request. Cryptographic schemes to accomplish this aim are known as *private information retrieval* (PIR) schemes. More formally, let Bob be a user, and let Alice be a server that maintains a database containing items $\alpha = \{a_1, a_2, \ldots, a_n\}$; for instance, Alice may represent an advertiser, and α may represent the collection of advertisements held by Alice. The aim of a PIR scheme is to enable Bob to retrive an element $a_r \in \alpha$ of his choice from Alice in such a way that Alice learns no information about r. Of course, this may be accomplished trivially by having Alice send all of A

to Bob. As shown in [?], however, a PIR scheme may in fact be designed with $o(n)$ communication, in particular, n^ϵ communication for any $\epsilon > 0$ under the quadratic residuosity assumption. A number of improvements and variants have subsequently been proposed in the literature. Most recent of these is a scheme with $polylog(n)$ communication overhead [?]. None of the proposed PIR schemes, however, is practical for wide scale deployment.

In this paper, we consider a practical alternative to these proposed PIR schemes. In order to obtain improved communications and computational efficiency, we relax the common security model in two respects. First, in lieu of a single server (Alice), we assume a collection of servers among which a majority behave in an honest fashion. In other words, we make use of a *threshold* scheme. Second, we assume that requests from a large number of users may be batched, in which case it is acceptable for servers to learn what has been requested, but not by whom. In other words, in consonance with the "Crowds" principle, we permit full disclosure of aggregate information, but hide information regarding the requests of individual users. We refer to a scheme satisfying the PIR criteria with these two weakened assumptions as a *semi-private information retrieval* (SPIR) scheme. As we show, a SPIR scheme is capable of achieving communication overhead of $\Theta(1)$.

2 Definitions

Let $C_1, C_2, \ldots, C_k$ be a collection of consumers toward whom advertisements are to be directed. Let $P_1, P_2, \ldots, P_k$ be the respective profiles of these consumers. These profiles may contain any of a variety of pieces of information on the consumer, including standard demographic information such as age, sex, profession, annual income, etc., as well as other information such as recently visited URLs and search engine queries. Let us designate the set of possible consumer profiles by $\mathcal{P}$. We denote the advertiser by A, and let $\alpha = \{a_1, a_2, \ldots, a_n\}$ be the set of advertisements that A seeks to disseminate. The advertiser chooses a *negotiant function* $f_\alpha : \mathcal{P} \rightarrow Z_n$. This function takes the profile of a consumer as input and outputs a choice of advertisement from among α. As an example, f might derive a list of the most common words in URLs visited by the user and seek to match these to descriptors associated with the ads in α. Of course, it is possible to include additional inputs to f, such as the current date, or the list of advertisements already sent to the consumer. We leave these choices to the imagination of the reader.

In a *basic* negociant protocol, a consumer C_i downloads f from the advertiser A. He obtains $f(P_i)$ from A through an interactive protocol. In a *group* negociant protocol, we assume that A is represented by a collection of servers $S_1, S_2, \ldots, S_m$ who share a *bulletin board.* This is a piece of shared memory with authenticated appenditive write access, and is accessible by any server or consumer. Consumers post ad requests to the bulletin board in some form until a triggering event occurs. Servers then initiate communication with consumers and dispense ads to them. The following is a list of definitions and properties useful in describing negociant protocols.

Let l be a security parameter for a negociant protocol. We say that a function $h(l)$ is *negligible* if for any polynomial p, there exists a value d such that for $l \geq d$, we have $h(l) < 1/|p(l)|$. Otherwise, we say that h is non-negligible.

We say that a negociant protocol has *profile privacy* if the following holds for every consumer C_i. Let A_1 be an polynomial-time algorithm that takes as input the transcript of the negociant protocol and has access to all information possessed by A. Let A_2 be a polynomial-time algorithm that has access to all information possessed by A and also to $f(P_i)$. Let b be a

Targeted Advertising... And Privacy Too

Ari Juels[1]

RSA Laboratories
Bedford, MA 01730, USA
E-mail: ajuels@rsasecurity.com

Abstract.

1 Introduction

In February 2000, a major Web advertising firm known as DoubleClick touched off a furore in the press with the announcement that it would integrate offline information about consumers into its existing database of online information derived from surveillance of consumer Web surfing. This announcement by DoubleClick came on the heels of a number of other articles in the popular press regarding abuse of private consumer information. A week earlier, a report released by the California HealthCare Foundation alleged that a number of health-related Web sites were violating their own stated privacy policies and divulging sensitive information about customers to third parties. The next day, Reuters reported that two suits for privacy invasion and an investigation by the Federal Trade Commission were pending against Amazon.com and its subsidiary Alexa. While consumer and privacy advocacy groups vigorously decry such abuses, advertisers defend their policy of harvesting and exploiting demographic information by highlighting the benefits of targeted advertising. Consumers, they maintain, are more likely to find interest in advertising tailored to their own preferences, and such advertising consequently leads to greater consumer market efficiency. The United States government has addresed the issue by promoting a policy of industry self-regulation, leading to friction with the European Union, which has sought more stringent consumer privacy guarantees.

In this paper, we show that targeted advertising and consumer privacy need not in fact be conflicting aims. We describe a simple, practical technical solution that enables sophisticated use of detailed consumer profiles for the purposes of targeting advertisements, but protects these profiles from disclosure to advertisers or hostile third parties. Somewhat surprisingly, the most basic embodiments of our idea do not even require use of cryptographic techniques.

The underlying idea is quite simple. Rather than gathering information about a consumer in order to decide which advertisements to send her, an advertiser makes use of a client-side software module called a *negociant*. The *negociant* serves a dual purpose: It acts as a client-side proxy to protect user information, and also directs the targeting of advertisements. The negociant requests advertisements from the advertiser that are tailored to the profile provided by the user. The advertiser can control the palette of advertisements available to the negociant, as well as the process by which it decides which ads to request. At the same time, the advertiser learns no information about the consumer profile beyond which advertisements the negociant requested. In more sophisticated variants, the negociant is able to participate in a protocol whereby the advertiser does not even learn what ads a given user has requested, but only sees ad requests in the aggregate. The end result is that the advertiser is able to target ads with a high degree of sophistication, and also to gather information on ad display rates, all without learning significant information about individual consumer profiles.

Of course, some restriction must be placed on advertiser control of negociants. Otherwise, the advertiser can manipulate them so as to extract detailed profile information from individual consumers. The fact that negociants may be viewed and controlled by users helps offset this vulnerability, as we discuss in the body of the paper. An additional point of concern in our proposal is a restriction that it places on advertisers. With the use of negociants, advertisers cannot correlate profile information among users, as is possible when consumer profiles are collected in a central location. This drawback should be partially offset by the fact that the negociant may safely and privately gain access to a great deal of sensitive information that would otherwise not be available to advertisers. Nonetheless, we propose some strategies for addressing this limitation in a manner that preserves consumer privacy.

1.1 Previous Work

A negociant may be viewed as a client-side software proxy. The approach of using proxies as a means of protecting consumer privacy is a well established one, and has seen application by both research and commercial ventures. These efforts, however, have typically involved proxy *servers*, i.e., servers that act as intermediaries between consumers and Web sites. (Of course, it is possible to install a negociant on a proxy server, rather than the client. This requires, however, an embodiment of trust in the proxy server.) A number of proposals...

Consumer privacy in P3P – language of privacy

In the stronger variant of our scheme, we wish to ensure against advertisers learning which advertisements have been requested by which users. In principle, it is possible for a consumer to request an advertisement from a server such that the server learns no information about the request. Cryptographic schemes to accomplish this aim are known as *private information retrieval* (PIR) schemes. More formally, let Bob be a user, and let Alice be a server that maintains a database containing items $\alpha = \{a_1, a_2, \ldots, a_n\}$; for instance, Alice may represent an advertiser, and α may represent the collection of advertisements held by Alice. The aim of a PIR scheme is to enable Bob to retrive an element $a_r \in \alpha$ of his choice from Alice in such a way that Alice learns no information about r. Of course, this may be accomplished trivially by having Alice send all of A

to Bob. As shown in [?], however, a PIR scheme may in fact be designed with $o(n)$ communication, in particular, n^ϵ communication for any $\epsilon > 0$ under the quadratic residuosity assumption. A number of improvements and variants have subsequently been proposed in the literature. Most recent of these is a scheme with $polylog(n)$ communication overhead [?]. None of the proposed PIR schemes, however, is practical for wide scale deployment.

In this paper, we consider a practical alternative to these proposed PIR schemes. In order to obtain improved communications and computational efficiency, we consider two relaxations of the common security model. First, in lieu of a single server (Alice), we assume a collection of servers among which a majority behave in an honest fashion. We refer to this as a *threshold* PIR scheme. As we show, a threshold PIR scheme is capable of achieving communication overhead of $\Theta(1)$ per consumer request under appropriate computational assumptions. As a second relaxation, we consider a scenario in which requests from a large number of users may be batched, in which case it is acceptable for servers to learn what has been requested, but not by whom. In other words, in consonance with the "Crowds" principle, we permit full disclosure of aggregate information, but hide information regarding the requests of individual users. We refer to a threshold PIR scheme with this latter property as a *semi-private information retrieval* (SPIR) scheme. A SPIR scheme, in addition to achieving communication overhead of $\Theta(1)$, is computationally quite efficient, involving $\Theta(1)$ basic cryptographic operations per item per server.

2 Preliminaries

2.1 Building blocks

We begin by introducing some of the cryptographic primitives used in the more sophisticated variants of our protocol. Readers with extensive familiarity of the basic cryptographic literature may wish to skip to Section 2.2.

El Gamal cryptosystem: A convenient basis for many election schemes, including those we employ here, is the El Gamal cryptosystem [?,?]. Encryption takes place over a group G_q of prime order q. Typically, G_q is taken to be a subgroup of Z_p^*, where $q|p-1$, but alternatives are possible; for example, G_q may be the group of points of an elliptic curve over a finite field.[1]

Let g be a generator of G_q; this generator is typically regarded as a system parameter, since it may correspond to multiple key pairs. The private encryption key consists of an integer $x \in_U Z_q$, where $\in_U$ denotes uniform random selection. The corresponding public key is defined to be $y = g^x$. To encrypt a message $m \in G_q$, we select $a \in_U$, and compute the ciphertext $(\alpha, \beta) = (my^a, g^a)$. To decrypt this ciphertext using the private key x, we compute $\alpha/\beta^x = my^a/(g^a)^x = m$.

[1] Most commonly, we let $p = 2q + 1$, and we let G_q be the set of quadratic residues in Z_p^*. In this setting, plaintexts not in $\mathcal{G}_q$ can be mapped onto $\mathcal{G}_q$ by appropriate forcing of the LeGendre symbol, e.g., inversion of the associated integer sign.

We assume a consistent choice of g as a generator for all instantiations of the El Gamal cryptosystem in this paper.

The El Gamal cryptosystem is *semantically secure* under the Decision Diffie-Hellman assumption over G_q [?,?]. Informally, this means that an attacker who selects message pair (m_0, m_1) is unable to distinguish between encryptions of these two messages with probability significantly greater than 1/2. (See [?] for futher details.)

Let $(\alpha_0\alpha_1, \beta_0\beta_1) = (\alpha_0, \beta_0) \otimes (\alpha_1, \beta_1)$. Another useful property of the El Gamal cryptosystem is the fact that it possesses a *homomorphism* under the operator $\otimes$. In particular, observe that if (α_0, β_0) and (α_1, β_1) represent ciphertexts corresponding to plaintexts m_0 and m_1 respectively, then $(\alpha_0, \beta_0) \otimes (\alpha_1, \beta_1)$ represents an encryption of the plaintext $m_0 m_1$. A consequence of this homomorphic property is that it is possible, using knowledge of the public key alone, to derive a random *re-encryption* (α', β') of a given ciphertext (α, β). This is accomplished by computing $(\alpha', \beta') = (\alpha, \beta) \otimes (\gamma, \delta)$, where (γ, δ) represents an encryption of the plaintext value 1. It is possible to prove quite efficiently in zero-knowledge that (α', β') represents a valid re-encryption of (α, β) using a variant of the Schnorr proof of knowledge protocol [?]. This proof may also be made non-interactive. See [?] for an overview.

Bulletin Board: Our proposed schemes with multiple players or servers assume the availability of a *bulletin board*. This may be viewed as a piece of memory which any player may view and to which all players have appendative write access. A bulletin board may be realized as a public broadcast channel, achievable through Byzantine agreement or through some appropriate physical assumption. See [?,?] for description of a practical implemenation of this primitive. Postings to a bulletin board may be made authenticable, i.e., their source may be securely validated, through use of such mechanisms as digital signaturess.

Mix networks: An important building block in one of our protocols is a *mix network*. This is a primitive executed by a collection of servers $S_1, S_2, \ldots, S_m$ with an appropriately shared public/private key pair (y, x). Let $E_y[M]$ represent the encryption under public key y of message M in a probabilistic public-key cryptosystem E. A mix network takes as input a vector of ciphertexts $V = \{E_y[M_1], E_y[M_2], \ldots, E_y[M_n]\}$. Output from the mix network is the vector $V' = \{E_y[M_\sigma(1)], E_y[M_\sigma(2)], \ldots, E_y[M_\sigma(n)]\}$, where σ is a random permutation on n elements. A mix scheme is said to be *robust* if, given a static adversary with active control of a minority coalition of servers, V' represents a valid permutation and re-encryption of ciphertexts in V with overwhelming probability. A mix scheme is said to be *private* if, given valid output V', for any $i \in \{1, 2, \ldots, n\}$, an adversary with passive control of at most $m-1$ servers cannot determine $\sigma^{-1}(i)$ with probability non-negligibly larger than $1/n$.

Mix servers were introduced by Chaum as a basic primative for privacy. In his simple formulation, each server S_i takes the output V_i of the previous server and simply permutes and re-encrypts the ciphertexts therein. While this scheme is private, it is not robust. A robust mix network was introduced by.

The most efficient to date is the construction of Jakobsson, which requires $O(n)$ computation and communication per server. The scheme makes use of the El Gamal cryptosystem. For further details and formal definitions, the reader is referred to [?]. Given its robustness and efficiency on large batches, the Jakobsson construction is probably most appropriate for our schemes. Given, however, that robustness is not of critical importance in these schemes, even the mix network proposed by Chaum may often be appropriate.

There are many variations on mix networks. For example, V may be a vector of k-tuples of ciphertexts [?]. Additionally, a mix network may take either ciphertexts or plaintexts as inputs and likewise output either plaintexts or ciphertexts. We employ a variety of such operations in our protocols, and do not describe implementation details.

Robust threshold El Gamal cryptosystem: A robust (j, m)-threshold cryptosystem is one in which a private key is held distributively by m players in such a way that a ciphertext may only be decrypted by j of them; decryption takes place without leakage of information about the private key. (See, e.g., [?] for a survey.) The Pedersen protocol [?,?] may be used as a basis for key generation in such a scheme for El Gamal, although see [?] for a caveat. For a description of a corresponding decryption algorithm, see, e.g., [?]. These algorithms are, in a practical sense, quite efficient.

Distributed ciphertext key transformation (DIKT) This is essentially a variant on robust threshold El Gamal decryption. The protocol is executed by servers $S_1, S_2, \ldots, S_m$ holding El Gamal secret key x in an appropriate distributed fashion. For simplicity of presentation, we assume here that x is shared additively among the servers. In particular, we assume that server S_i holds public/private share $(x_i, y_i = g^{x_i})$ such that $x = \sum_{i=1}^{m} x_i$. (Robustness is achievable by having a second level sharing, i.e., a sharing of shares.) Our protocol can be easily extended to more natural sharing schemes, such as threshold schemes based on Shamir secret sharing. We omit discussion of relevant threshold key generation and management issues, and instead refer the reader to, e.g., [?]. Let y be the public key corresponding to x.

The DIKT protocol takes as input an El Gamal public key y', as well as a ciphertext $(\alpha, \beta) = E_y[M]$. The output of the protocol is $(\alpha', \beta') = E_{y'}[M]$. The protocol is said to be *robust* if, given an adversary with active control of a static minority coalition of servers, the output of the protocol is correct. It is said to be *private* if such an adversary learns no information about M. In particular, we require, given the presence of such an adversary, that M be semantically secure under this protocol, in a sense analogous to that for the underlying El Gamal cryptosystem. As in the other primitives described in this paper, we assume the availability of a bulletin board.

We propose a protocol in which each server S_i does the following.

1. S_i chooses $w \in_U Z_q$.
2. S_i computes $(\gamma_i, \delta_i) = (\beta^{x_i} y'^{w}, g'^{w})$ and posts it to the bulletin board.

3. S_i posts a proof, relative to her public key y_i, that (γ_i, δ_i) is correctly formulated.

This last step may be accomplished by having S_i prove knowledge of exponents d_1 and d_2 such that $\gamma_i = \beta^{d_1} y'^{d_2}$ and $\delta_i = g'^{d_2}$ and $y_i = x_i^{d_1}$. This may be done efficiently and in either interactive or non-interactive zero knowledge using protocols described in, e.g., [?,?].

If any server proof is incorrect, that server is expelled from the coalition, and the protocol is performed again with the remaining servers. Once all servers have correctly posted their partial computations, the servers jointly compute $(\alpha', \beta') = (\alpha / \prod_{i=1}^{m} \gamma_i, \beta / \prod_{i=1}^{m} \delta_i)$. The DIKT scheme may be shown to be robust and private relative to the Decision Diffie-Hellman assumption.

Distributed plaintext equality test This is a protocol whereby, given El Gamal ciphertexts (α, β) and (α', β'), a collection of servers determine whether the underlying plaintexts are identical. The protocol may be performed in a robust manner, and in such a way that no additional information is revealed. See [?] for an efficient protocol construction and proofs of security. We write $(\alpha, \beta) \approx (\alpha', \beta')$ to denote equality of underlying plaintexts.

2.2 Model and definitions for our scheme

Let $C_1, C_2, \ldots, C_k$ be a collection of consumers toward whom advertisements are to be directed. Let $P_1, P_2, \ldots, P_k$ be the respective profiles of these consumers. These profiles may contain any of a variety of pieces of information on the consumer, including standard demographic information such as age, sex, profession, annual income, etc., as well as other information such as recently visited URLs and search engine queries. Let us designate the set of possible consumer profiles by $\mathcal{P}$. We denote the advertiser by A, and let $\alpha = \{a_1, a_2, \ldots, a_n\}$ be the set of advertisements that A seeks to disseminate. The advertiser chooses a *negotiant function* $f_\alpha : \mathcal{P} \rightarrow Z_n$. This function takes the profile of a consumer as input and outputs a choice of advertisement from among α. As an example, f might derive a list of the most common words in URLs visited by the user and seek to match these to descriptors associated with the ads in α. Of course, it is possible to include additional inputs to f, such as the current date, or the list of advertisements already sent to the consumer. We leave these choices to the imagination of the reader. We assume that A is represented by a collection of servers $S_1, S_2, \ldots, S_m$ who share a bulletin board. All consumers post ad requests to the bulletin board. Servers then initiate communication with consumers and dispense ads to them. The following is a list of definitions and properties useful in describing negociant protocols.

Let l be an appropriately defined security parameter. We say that a function $q(l)$ is *negligible* if for any polynomial p, there exists a value d such that for $l \geq d$, we have $q(l) < 1/|p(l)|$. Otherwise, we say that q is *non-negligible*. We say that a probability function $q(l)$ is *overwhelming* if $1 - q(l)$ is negligible.

Let A_1 be a polynomial-time adversary that actively controls a static minority set of servers, or, if there is only one server, controls that single server. In other words, let us suppose that A_1 controls $max(\lfloor m/2 \rfloor, 1)$ servers. Consider the following experiment. Let us assume without loss of generality that A_1 does not control consumer C_1. A_1 chooses a pair of profiles $\tilde{P}_0, \tilde{P}_1 \in \mathcal{P}$. A bit $b \in \{0, 1\}$ is selected at random and P_1 is set to $\tilde{P}_b$. Now the protocol is run. We say that the protocol has *full privacy* if for any adversary A_1, it is the case that $\mathsf{pr}[A_1 \text{ outputs } b] - 1/2$ is negligible, where the probability is taken over the coin flips of all participants. This definition states informally that the protocol transcript reveals no significant information about P_1.

Now let us modify the experiment slightly and consider a polynomial-time algorithm A_2 identical to A_1, except that A_2 is provided only with the input $f(P_1), f(P_2), \ldots, f(P_n)$ in a random order. We say that a negociant protocol has *profile privacy* if for any A_1 and any $P_2, P_3, \ldots, P_n$, there exists an A_2 such that $\mathsf{pr}[A_1 \text{ outputs } b] - \mathsf{pr}[A_2 \text{ outputs } b]$ is negligible if all consumers adhere to the correct protocol. Again, probabilities are taken over the coin flips of all participants. In other words, the protocol transcript reveals no signficant information about P_1 other than that revealed by the aggregate ad requests of the participating consumers. Note that when $m = 1$, the property of profile privacy means that an advertiser learns only the ad requests of a consumer. When $m > 1$, the property implies that an advertiser learns only the agregate ad requests of a group of consumers.

We say that a negociant protocol is *aggregate transparent* if any server can determine $f(P_1), f(P_2), \ldots, f(P_n)$ with overwhelming probability. In real-world advertising scenarios, it is important that a protocol be aggregate transparent, as the clients of advertisers wish to konw how many times their ads have been displayed.

3 Some Negociant Schemes

We now present a small spectrum of negociant schemes with different properties and resource costs.

3.1 Naive PIR scheme

We present this simple scheme as a conceptual introduction. Here, requests are directed from a single consumer C with profile P to a single server S. (Thus the scheme may be modeled by $m = n = 1$.) The scheme is this: The server sends all of α to C, who then views ad $a_{f(P)}$.

Clearly, this scheme enjoys full privacy. The chief drawback is the $O(n)$ communication cost. Another drawback is the fact that the scheme is not aggregate transparent. Nonetheless, given a limited base of advertisements and good bandwidth, and if advertisers are satisfied with recording click-through rates, this scheme may be useable in certain practical scenarios.

3.2 Direct request scheme

This is another conceptually simple scheme involving a one-on-one consumer and server interaction. In this scheme, C simply sends $f(P)$ to S, who returns $a_{f(P)}$. This scheme enjoys profile privacy and has communication and computation costs $\Theta(1)$. Despite (or because of) its simplicity, it is from a practical standpoint perhaps the most appealing of the schemes proposed here.

3.3 SPIR scheme

We now show how to invoke some of the cryptographic apparatus described above in order to achieve a semi-private information retrival (SPIR) scheme. Given database $\alpha = \{a_1, a_2, \ldots, a_n\}$, the goal is for a collection of consumers $C_1, C_2, \ldots, C_k$ to retrieve respective elements $\{a_{r_i}\}$ in such a way that the database servers learn requests only in the aggregate. Of course, our aim here is to apply this scheme to the retrieval of advertisements, and we shall present it in this context. In other words, we assume that $r_i = f(P_i)$, i.e., users are consumers seeking to retrieve advertisements. As above, we assume a public/private El Gamal key pair (y, x) held in an appropriate distributed manner by servers $S_1, S_2, \ldots, S_m$. We also assume that each consumer C_i has a public/private El Gamal key pair (y_{C_i}, x_{C_i}). The scheme is as follows.

1. Each consumer C_i computes $r_i = f(P_i)$ and posts the pair $(E_y[r_i], i)$ to the bulletin board. Let $V_1 = \{E_y[r_i], i\}_{i=1}^k$ be a vector of ciphertext/plaintext pairs accumulated when all consumers have posted their requests.
2. Servers apply a mix network to V_1 to obtain V_2, where V_2 is a vector of pairs $\{(r_{\sigma(i)}, E_y[\sigma_1(i)])\}_{i=1}^k$ for random, unknown permutation σ_1.
3. Servers replace each integer r_j in V_2 with a_{r_j}. Call the resulting vector V_2'.
4. Servers apply a mix network to V_2' to obtain a vector V_3, where V_3 is a vector of pairs $\{(E_y[a_{r_{\sigma_2(i)}}], i)\}_{i=1}^k$, and σ_2 is an unknown random permutation.
5. Let $(E_y[a_j], i)$ be an element in V_3. For each pair, the servers apply DIKT to obtain $(E_{y_{C_i}}[a_j], i)$. Let the resulting vector be V_4.
6. For each element $(E_{y_{C_i}}[a_j], i)$ in V_4, servers send $E_{y_{C_i}}[a_j]$ to C_i.
7. Consumers decrypt their respective ciphertexts.

It may be shown that this is a SPIR scheme, i.e., enjoys profile privacy, relative to the Decision Diffie-Hellman assumption. Assuming that a public key operation in G_q incurs cost $\Theta(l^3)$, where it will be recalled that l is a security parameter, the computational costs of the scheme are $\Theta(l^3)$ per element per server. The communication costs of the scheme are $\Theta(1)$. With some efficiency enhancements, we believe that this scheme may be implemented in a fairly practical manner.

Bulk encryption We assume here that an advertisement may be represented as a single ciphertext. Of course, in reality, it is impractical to use ads small enough or a group G_q large enough to support this assumption. We may, however, represent an advertisement as a sequence of associated ciphertexts. Alternatively, we may use an enveloping scheme, and represent a encryption of M as $\tilde{E}_y[M] = (\gamma, \delta)$, where $\gamma = \{E_y[\kappa_1], E_y[\kappa_2], \ldots, E_y[\kappa_z]\}$ and $\delta = \epsilon_{\kappa_z}\epsilon_{\kappa_{z-1}}\epsilon_1[M]$. Here, $\epsilon_\kappa[M]$ represents a symmetric-key encryption of M, where $\kappa \in K$ is a key from keyspace K. To re-encrypt $\tilde{E}_y[M]$ as (γ', δ'), a player does the following:

1. Re-encrypt all ciphertexts in γ.
2. Select $\kappa_{z+1} \in_U K$.
3. Append $E_y[\kappa_{z+1}]$ to γ to obtain γ'.
4. Compute δ' as $\epsilon_{\kappa_{z+1}}[\delta]$.

There are two major drawbacks to this scheme. First, the size of a ciphertext, as well as the computational cost of re-encryption, grows linearly in the number of re-encryptions z. While this leads to poor asymptotic performance, it may substantially enhance the performance in practice, particularly when m is small and ad sizes are large. A second drawback is that it is unclear how to achieve robustness in an efficient way in a mix network employing such a scheme. We submit, however, that robustness is a less important consideration than privacy in our schemes.

Reducing public-key operation costs It should be noted that all of the costly operations in our scheme involve exponentiations in G_q. These may be made quite inexpensive through the use of pre-processing or addition chains. Thus, assuming 100,000 elements, the total cost is roughly that of...

3.4 Threshold PIR

The SPIR scheme described above can be strengthened with a few extra steps, and at the expense of additional computational overhead, into a threshold PIR scheme. The idea is to perform a blind lookup of consumer ad requests. This is accomplished by mixing ads and then invoking the distributed plaintext equality test described in Section 2.1. We present the protocol here.

1. Each consumer C_i computes $r_i = f(P_i)$ and posts the pair $(E_y[r_i], i)$ to the bulletin board. Let $V_1 = \{E_y[r_i], i\}_{i=1}^{k}$ be a vector of ciphertext/plaintext pairs accumulated when all consumers have posted their requests.
2. For each pair $(E_y[r_i], i)$, servers do the following:
   (a) Servers construct a vector U_1 of pairs $(j, E_y[a_j])$.
   (b) Servers mix U_1 to obtain a vector U_2 in which an entry takes the form $(E_y[j], E_y[a_j])$.
   (c) For each j, until a match is found, servers perform a distributed plaintext equality test to see whether $E_y[j] \approx E_y[r_i]$.
   (d) When a match is found, servers replace $E_y[r_i]$ with $E_y[a_j] = E_y[a_{r_i}]$.
3. Let V_2 be the resulting vector, and let $(E_y[a_{r_i}], i)$ be an element therein. For each such pair, the servers apply DIKT to obtain $(E_{y_{C_i}}[a_j], i)$. Let the resulting vector be V_3.
4. For each element $(E_{y_{C_i}}[a_j], i)$ in V_3, servers send $E_{y_{C_i}}[a_j]$ to C_i.
5. Consumers decrypt their respective ciphertexts.